Sign in

Google Web Images Groups News Froogle Local[New!] more »

memory address endian conversion Search Advanced Search Preferences

O Search the Web ◉ Search English and German pages

Web Results **1** - **10** of about **204,000 English and German** pages for **memory address endian conversion**. (**0.18** seconds)

[PDF] Big Endian to Little Endian Data Conversion
File Format: PDF/Adobe Acrobat - View as HTML
the lowest **memory address**. The Intel Pentium. TM. and Compaq/Dec ... Often, based on the application, it is required to **convert** Big **Endian** ...
www.pericom.com/pdf/applications/AB034.pdf - Similar pages

BYTE.com
Thus, the **conversion** of a **memory** block from one style of endianness to the other ... This ability to handle bi-**endian address** modes makes the PowerPC ...
www.byte.com/art/9509/sec12/art1.htm - 34k - Cached - Similar pages

Basic concepts on Endianness - The Code Project - C++ / MFC
... at the lowest **memory address**, which is also the **address** of the larger field. ... The following example shows how an **Endian conversion** function could be ...
www.codeproject.com/cpp/**endian**ness.asp - 52k - Cached - Similar pages

DAV's Endian FAQ
This theorem is the basis of computer **memory addressing**. ... Furthermore, by having **endian-conversion** for > DMA data programmable through the DMA descriptor ...
www.rdrop.com/~cary/html/**endian**_faq.html - 107k - Cached - Similar pages

ITworld.com - LINUX TIPS AND TRICKS - Byte Ordering
Big **endian** machines store the most significant byte at the lowest **memory address**. The remaining bytes occupy higher **memory addresses**, in ascending order. ...
www.itworld.com/nl/lnx_tip/04062001/ - 39k - Cached - Similar pages

NETBOOK - Q & A on Data Typing IP Addresses
In our 636 project, we never **convert** the IPaddr between the network order and ... highest **address**(I could not remember it is big **endian** or little **endian**). ...
www.netbook.cs.purdue.edu/othrpags/qanda49.htm - 5k - Cached - Similar pages

Lab 4 - Serial Peripheral Interface Analog to Digital Conversion
In this lab you will acquire data from an analog to digital **converter** chip using ... On big-**endian** machines the lower **memory address** (sc.c[0]) holds the ...
physics.usask.ca/~angie/ep414/labmanual/ep414labmanual/node5.html - 20k - Cached - Similar pages

Unix network programming: a practical approach -- part 2
... and little-**endian** is in which direction increasing **memory addresses** go. ... If our host is little-**endian**, the **conversion** functions will **convert** values ...
bdn.borland.com/article/0,1410,26267,00.html - 25k - Cached - Similar pages

Byte Alignment and Ordering in Message Definition
... microprocessors do not use the **address** lines A1 and A0 for **addressing memory**). ... Routines to **convert** between big-**endian** and little-**endian** formats are ...
www.eventhelix.com/RealtimeMantra/ByteAlignmentAndOrdering.htm - 27k - Cached - Similar pages

September 1995 / Core Technologies / Endian Issues
Thus, the **conversion** of a **memory** block from one style of endianness to the ... Each data item has the same **address** in both big- and little-**endian** schemes. ...
www.cs.utexas.edu/users/chris/cs310/f98/notes/**endian**.html - 26k - Cached - Similar pages

Try your search again on Google Book Search

Gooooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

memory address endian conversic Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Terms used **endian memory address** Found **152** of **167,655**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 152 Result page: **1** 2 3 4 5 6 7 8 next

Relevance scale

1 Communication and devices: HAIL: a language for easy and correct device access

Jun Sun, Wanghong Yuan, Mahesh Kallahalla, Nayeem Islam

September 2005 **Proceedings of the 5th ACM international conference on Embedded software EMSOFT '05**

Publisher: ACM Press

Full text available: pdf(477.40 KB) Additional Information: full citation, abstract, references, index terms

It is difficult to write device drivers. One factor is that writing low-level code for accessing devices and manipulating their registers is tedious and error-prone. For many system-on-chip based systems, buggy hardware, imprecise documentation, and code reuse worsen the situation further. This paper presents *HAIL (Hardware Access Interface Language)*, a language-based approach to simplify device access programming and generate error checking code against bugs in software, hardware, and do ...

Keywords: automatic code generation, device drivers, domain-specific languages, embedded systems, invariant specification and verification, register access, software reuse, system-on-chip

2 Alpha AXP architecture

Richard L. Sites

February 1993 **Communications of the ACM**, Volume 36 Issue 2

Publisher: ACM Press

Full text available: pdf(4.62 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: Alpha AXP chip

3 Endian-safe record representation clauses for Ada programs

Mike Mardis

December 1999 **ACM SIGAda Ada Letters**, Volume XIX Issue 4

Publisher: ACM Press

Full text available: pdf(294.44 KB) Additional Information: full citation, index terms

4 Using the Alfa-1 simulated processor for educational purposes

Gabriel A. Wainer, Sergio Daicz, Luis F. De Simoni, Demian Wassermann

December 2001 **Journal on Educational Resources in Computing (JERIC)**, Volume 1 Issue 4

Publisher: ACM Press

Full text available: pdf(238.65 KB) Additional Information: full citation, abstract, references, index terms

Alfa-1 is a simulated computer designed for computer organization courses. Alfa-1 and its accompanying toolkit allow students to acquire practical insights into developing hardware by extending existing components. The DEVS formalism is used to model individual components and to integrate them into a hierarchy that describes the detailed behavior of different levels of a computer's architecture. We introduce Alfa-1 and the toolkit, show how to extend existing components, and describe how ...

Keywords: DEVS formalism, modeling computer architectures, systems specification

5 The X window system

Robert W. Scheifler, Jim Gettys

April 1986 **ACM Transactions on Graphics (TOG)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(2.76 MB) Additional Information: full citation, abstract, references, citings, index terms, review

An overview of the X Window System is presented, focusing on the system substrate and the low-level facilities provided to build applications and to manage the desktop. The system provides high-performance, high-level, device-independent graphics. A hierarchy of resizable, overlapping windows allows a wide variety of application and user interfaces to be built easily. Network-transparent access to the display provides an important degree of functional separation, without significantly affec ...

6 Eliminating redundancies with a "composition with adaptation" meta-programming technique

Stan Jarzabek, Li Shubiao

September 2003 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11**, Volume 28 Issue 5

Publisher: ACM Press

Full text available: pdf(297.50 KB) Additional Information: full citation, abstract, references, citings, index terms

Redundant code obstructs program understanding and contributes to high maintenance costs. While most experts agree on that, opinions - on how serious the problem of redundancies really is and how to tackle it - differ. In this paper, we present the study of redundancies in the Java Buffer library, JDK 1.4.1, which was recently released by Sun. We found that at least 68% of code in the Buffer library is redundant in the sense that it recurs in many classes in the same or slightly modified form. W ...

Keywords: class libraries, generative programming, meta-programming, object-oriented methods

7 The KScalar simulator

J. C. Moure, Dolores I. Rexachs, Emilio Luque

March 2002 **Journal on Educational Resources in Computing (JERIC)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(493.35 KB) Additional Information: full citation, abstract, references, index terms

Modern processors increase their performance with complex microarchitectural mechanisms, which makes them more and more difficult to understand and evaluate. KScalar is a graphical simulation tool that facilitates the study of such processors. It allows students to analyze the performance behavior of a wide range of processor microarchitectures: from a very simple in-order, scalar pipeline, to a detailed out-of-order, superscalar pipeline with non-blocking caches, speculative execution, and comp ...

Keywords: Education, pipelined processor simulator

8 Reconfigurable, retargetable bignums: a case study in efficient, portable Lisp system building

Jon L. White

August 1986 **Proceedings of the 1986 ACM conference on LISP and functional programming**

Publisher: ACM Press

Full text available: pdf(1.70 MB) Additional Information: full citation, references, citings

9 An unconventional proposal: using the x86 architecture as the ubiquitous virtual standard architecture

Jochen Liedtke, Nayeem Islam, Trent Jaeger, Vsevolod Panteleenko, Yoonho Park

September 1998 **Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications**

Publisher: ACM Press

Full text available: pdf(410.74 KB) Additional Information: full citation, index terms

10 The Parallel Protocol Engine

Matthias Kaiserswerth

December 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1 Issue 6

Publisher: IEEE Press

Full text available: pdf(1.65 MB) Additional Information: full citation, references, citings, index terms, review

11 The i860™ 64-bit supercomputing microprocessor

L. Kohn, N. Margulis

August 1989 **Proceedings of the 1989 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available: pdf(575.98 KB) Additional Information: full citation, abstract, references, citings, index terms

The Intel i860TM processor is a RISC-based microprocessor incorporating a RISC core with memory management, a floating point unit, and caches on a single chip. The 1,000,000 transistors allow a single chip implementation with highly optimized interunit communication and wide internal data buses. The parallelism and pipelining between the execution units, and the innovative cache management techniques are under explicit control of software. Vectorizable applications can ...

12 The Clipper processor: instruction set architecture and implementation

W. Hollingsworth, H. Sachs, A. J. Smith

February 1989 **Communications of the ACM**, Volume 32 Issue 2

Publisher: ACM Press

Full text available: pdf(4.67 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Intergraph's CLIPPER microprocessor is a high performance, three chip module that implements a new instruction set architecture designed for convenient programmability, broad functionality, and easy future expansion.

13 Efficient and language-independent mobile programs

Ali-Reza Adl-Tabatabai, Geoff Langdale, Steven Lucco, Robert Wahbe

May 1996 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1996 conference on Programming language design and implementation PLDI '96**, Volume 31 Issue 5

Publisher: ACM Press

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper evaluates the design and implementation of Omniware: a safe, efficient, and language-independent system for executing mobile program modules. Previous approaches to implementing mobile code rely on either language semantics or abstract machine interpretation to enforce safety. In the former case, the mobile code system sacrifices universality to gain safety by dictating a particular source language or type system. In the latter case, the mobile code system sacrifices performance to ga ...

14 Four Dimensions of programming-language independence

Daniel J. Salomon

March 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 3

Publisher: ACM Press

Full text available: pdf(2.04 MB) Additional Information: full citation, abstract, index terms

The features of programming languages can be evaluated according to how they affect programming-language independence in four dimensions. The four dimensions are: 1) machine independence, 2) problem independence, 3) human independence, and 4) time independence. This paper presents a definition of *independence*, and shows how that definition applies to each of the dimensions. By organizing language features in this way, the strengths and weaknesses of many language designs can be identified ...

15 An introduction to the Beetle Forth virtual processor

Reuben R. Thomas
February 1997 **ACM SIGPLAN Notices**, Volume 32 Issue 2
Publisher: ACM Press
Full text available: pdf(414.59 KB) Additional Information: full citation, abstract, index terms

Beetle is a virtual processor designed for the Forth language. It uses a modified, byte-stream code designed for efficient execution which is binary portable between implementations. It has been implemented in C and assembler. The C implementation is completely machine independent with the exception of interactive input and output; the assembler version runs the supplied Forth compiler at up to half the speed of the corresponding native code compiler and generates more compact code. Beetle is de ...

16 Improving the efficiency of the OSI checksum calculation

K. Sklower
October 1989 **ACM SIGCOMM Computer Communication Review**, Volume 19 Issue 5
Publisher: ACM Press
Full text available: pdf(629.71 KB) Additional Information: full citation, abstract, citings, index terms

It is known that using larger byte--sizes to access memory usually results in faster computations of checksum algorithms. This paper proposes two different ways to use larger byte--sizes to improve the performance of the OSI checksum. First, an algorithm is presented that computes the 8--bit checksum using 16-bit integers. It is shown that this algorithm yields a 5 to 20 percent performance improvement on many architectures. Second, the benefits of expanding the basic computation unit of the OSI ...

17 A retargetable debugger

Norman Ramsey, David R. Hanson
July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation PLDI '92**, Volume 27 Issue 7
Publisher: ACM Press
Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

We are developing techniques for building retargetable debuggers. Our prototype, 1db, debugs C programs compiled for the MIPS R3000, Motorola 68020, SPARC, and VAX architectures. It can use a network to connect to faulty processes and can do cross-architecture debugging. 1db's total code size is about 16,000 lines, but it needs only 250–550 lines of machine-dependent code for each target. 1db owes its retargetability to three techniques: getting help from the compiler, usin ...

18 Implementing bit-addressing with specialization

Scott Draves
August 1997 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN international conference on Functional programming ICFP '97**, Volume 32 Issue 8
Publisher: ACM Press
Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, index terms

General media-processing programs are easily expressed with bit-addressing and variable-sized bit-fields. But the natural implementation of bit-addressing relies on dynamic shift offsets and repeated loads, resulting in slow execution. If the

code is specialized to the alignment of the data against word boundaries, the offsets become static and many repeated loads can be removed. We show how introducing modular arithmetic into an automatic compiler generator enables the transformation of a progr ...

19 The SimpleScalar tool set, version 2.0

Doug Burger, Todd M. Austin

June 1997 **ACM SIGARCH Computer Architecture News**, Volume 25 Issue 3

Publisher: ACM Press

Full text available: pdf(985.46 KB) Additional Information: full citation, abstract, citings, index terms

This document describes release 2.0 of the SimpleScalar tool set, a suite of free, publicly available simulation tools that offer both detailed and high-performance simulation of modern microprocessors. The new release offers more tools and capabilities, precompiled binaries, cleaner interfaces, better documentation, easier installation, improved portability, and higher performance. This paper contains a complete description of the tool set, including retrieval and installation instructions, a d ...

20 TetSplat Real-Time Rendering and Volume Clipping of Large Unstructured Tetrahedral Meshes

Ken Museth, Santiago Lombeyda

October 2004 **Proceedings of the conference on Visualization '04**

Publisher: IEEE Computer Society

Full text available: pdf(371.73 KB) Additional Information: full citation, abstract

We present a novel approach to interactive visualization and exploration of large unstructured tetrahedral meshes. These massive 3D meshes are used in mission-critical CFD and structural mechanics simulations, and typically sample multiple field values on several millions of unstructured grid points. Our method relies on the preprocessing of the tetrahedral mesh to partition it into non-convex boundaries and internal fragments that are subsequently encoded into compressed multi-resolution data r ...

Keywords: Large volumetric data, tetrahedral meshes, real-time visualization, point-based rendering, constructive solid geometry

Results 1 - 20 of 152 Result page: **1** 2 3 4 5 6 7 8 next

Search Session History

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Tue, 13 Dec 2005, 11:46:54 AM EST

Search Query Display

Run Search Reset

Recent Search Queries		Results
#1	((address<in>metadata) <and> (swizzling<in>metadata))	8
#2	((address<in>metadata) <and> (swizzling<in>metadata))<and> (endian<in>metadata)	0
#3	((endian<in>metadata) <and> (swizzling<in>metadata))	0

Clear Session History